

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Eric D. Fagerburg et al. Art Unit : 2157
Serial No.: 09/752,100 Examiner : Ramy M. Osman
Filed: December 28, 2000 Conf. No.: 8242
Title: REMOTELY CONTROLLING A UNIX-BASED SYSTEM

Mail Stop Appeal Brief - Patents

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

Applicant herewith files this brief on appeal under 37 CFR 41.37, thereby perfecting the notice of appeal which was originally filed on January 23, 2007.

The sections required by 37 CFR 41.37 follow.

(1) Real Party in Interest

This application is assigned of record to Intel Corporation who is hence the real party in interest.

(2) Related Appeals and Interferences

There are no known related appeals or interferences.

(3) Status of Claims

Claims 1-3, 5-13, 15-28, 31 and 32 are pending, with claims 1, 11, 21, 25, 27 and 31 being independent. Claims 5, 6, 15, 16, 26 and 32 have been indicated as allowable. Claims 1-3, 7-

13, 17-25, 27, 28 and 31 stand rejected, and the rejections of claims 1-3, 7-13, 17-25, 27, 28 and 31 are appealed herein.

(4) Status of Amendments

No claim amendments have been filed after the final rejection.

(5) Summary of Claimed Subject Matter

Independent claim 1 defines a method comprising:
replicating current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine (see, e.g., Specification at page 1, line 19 to page 3, line 18, page 7, lines 9-18, reference numerals 104, 122 and 126 in FIG. 1, and reference numeral 214 in FIG. 2); prompting a first user at the UNIX-based machine for permission for a second user at a machine remotely located from the UNIX-based machine to control the UNIX-based machine (see, e.g., Specification at page 1, line 19 to page 4, line 12, page 7, line 19 to page 9 line 14, reference numerals 102 and 106 in FIG. 1, and reference numerals 216-220 in FIG. 2); and if the first user grants permission, enabling the second user to use the UNIX-based machine through the machine remotely located from the UNIX-based machine (see, e.g., Specification at page 4, lines 12-21, page

10, lines 20-23, reference numerals 104 and 106 in FIG. 1, and reference numeral 232 in FIG. 2).

Independent claim 11 defines an article comprising: a machine-readable medium which stores machine-executable instructions (*see, e.g.*, Specification at page 6, line 22 to page 7, line 1, page 11, line 6 to page 12, line 11, and reference numerals 104 and 110 in FIG. 1), the instructions causing a machine to: replicate current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine (*see, e.g.*, Specification at page 1, line 19 to page 3, line 18, page 7, lines 9-18, reference numerals 104, 122 and 126 in FIG. 1, and reference numeral 214 in FIG. 2); prompt a first user at the UNIX-based machine for permission for a second user at a machine remotely located from the UNIX-based machine to control the UNIX-based machine (*see, e.g.*, Specification at page 1, line 19 to page 4, line 12, page 7, line 19 to page 9 line 14, reference numerals 102 and 106 in FIG. 1, and reference numerals 216-220 in FIG. 2); and if the first user grants permission, enable the second user to use the UNIX-based machine through the machine remotely-located from the UNIX-based machine (*see, e.g.*, Specification at page 4, lines 12-21, page 10, lines 20-23, reference numerals 104 and 106 in FIG. 1, and reference numeral 232 in FIG. 2).

Independent claim 21 defines a system comprising: a first device configured to run UNIX (*see, e.g.,* Specification at page 1, lines 19-21, and reference numeral 104 in FIG. 1); and a mechanism accessible by the first device and configured to run a process on the first device transparently to a user of the first device, the process configured to replicate current contents of a screen on the first device onto a new screen running in a background of the first device and prompt the user of the first device for permission for a remote user at a second device at a location remote from the first device to input instructions to the first device from the second device (*see, e.g.,* Specification at page 1, line 19 to page 11, line 3, and reference numerals 108 and 110 in FIG. 1).

Independent claim 25 defines a method comprising: replicating current contents of a display screen visible to a user on a UNIX-based device onto a new screen not visible on the display screen to the user (*see, e.g.,* Specification at page 1, line 19 to page 3, line 18, page 7, lines 1-18, reference numerals 104, 122 and 126 in FIG. 1, and reference numeral 214 in FIG. 2); inserting a prompt on the new screen to a user of the UNIX-based device to grant permission for a remote device at a location remote from the UNIX-based device to control the UNIX-based device (*see, e.g.,* Specification at page 1, line 19

to page 4, line 12, page 7, line 19 to page 9 line 14, reference numerals 102 and 106 in FIG. 1, and reference numerals 216-220 in FIG. 2); and replacing the current contents of the display screen with the new screen, the new screen visible to the user on the UNIX-based device (*see, e.g.*, Specification at page 8, lines 13-23, reference numeral 218 in FIG. 2, and reference numerals 104 and 126 in FIG. 4).

Independent claim 27 defines a method comprising:
replicating current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine (*see, e.g.*, Specification at page 1, line 19 to page 3, line 18, page 7, lines 9-18, reference numerals 104, 122 and 126 in FIG. 1, and reference numeral 214 in FIG. 2); adding to the new screen a text prompt prompting a first user at the UNIX-based machine for permission for a second user at a machine remotely-located from the UNIX-based machine to control the UNIX-based machine (*see, e.g.*, Specification at page 1, line 19 to page 4, line 12, page 7, line 19 to page 9 line 14, reference numerals 102 and 106 in FIG. 1, and reference numerals 216-220 in FIG. 2); and replacing the current contents of the screen with the new screen (*see, e.g.*, Specification at page 8, lines 13-23, reference numeral 218 in FIG. 2, and reference numerals 104 and 126 in FIG. 4).

Independent claim 31 defines a method comprising: receiving a request at a first programmable data processing machine for provision of remote operational control of the first machine to a second data processing machine, the first machine being a UNIX-based machine (*see, e.g.,* Specification at page 1, line 19 to page 3, line 18, page 5, line 16 to page 6, line 2, reference numerals 104, 122 and 126 in FIG. 1, and reference numeral 204 in FIG. 2); determining whether a user interface of the first machine is operating in a text mode (*see, e.g.,* Specification at page 6, lines 3-18, and reference numeral 206 in FIG. 2); and if the first machine user interface is operating in the text mode, allocating a virtual console running in a background of the first machine (*see, e.g.,* Specification at page 6, line 19 to page 7, line 8, and reference numeral 212 in FIG. 2), adding a prompt to the virtual console asking for authorization of the request (*see, e.g.,* Specification at page 7, line 19 to page 8, line 3, and reference numeral 216 in FIG. 2), and switching the virtual console to be a currently active console in the first machine (*see, e.g.,* Specification at page 8, lines 13-23, and reference numeral 218 in FIG. 2).

(6) Grounds of Rejection to be Reviewed on Appeal

I. Claims 1-3, 7-13, 17-25, 27, 28 and 31 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over DoubleVision 3.0 by Tridia in view of Muta (U.S. 6,286,003).

II. Claims 9 and 19 stand rejected under 25 U.S.C. 103(a) as allegedly being unpatentable over DoubleVision 3.0 by Tridia in view of Muta, in further view of Edwards (US 6,594,686).

(7) Argument

A *prima facie* case of obviousness has not been established. DoubleVision determines who can attach to a remote terminal based on the type of user and a set of predefined permissions, which are granted in advance using access control lists, private-users and private-groups databases, and a ".dvsc file". (See DoubleVision at section 6.6.) DoubleVision does not describe prompting a first user for permission for a second user to control a UNIX-based machine, as recited in independent claims 1, 11, 21, 25, 27 and 31.

In response to this argument, the Office has asserted, "it is broadly interpreted that a root user using DoubleVision grants permission to users via an access control list. The '.dvsc' file is a prompt for the root user to grant permission to a select number of authorized users." (See 03/10/2006 Office

Action at ¶ 4.) However, a ".dvsc" file is not a prompt.

Rather, it is a text file that a host user can create and edit as desired to grant permissions. (See DoubleVision at section 6.6.1, "USING A .DVSC FILE".) Thus, DoubleVision does not describe prompting a first user at a UNIX-based machine for permission for a second user at a machine remotely located from the UNIX-based machine to control the UNIX-based machine; and if the first user grants permission, enabling the second user to use the UNIX-based machine through the machine remotely located from the UNIX-based machine.

The Office has further asserted that, "the term 'prompt' is broad language and is interpreted to be a window on a computer screen allowing a user to input a command(s) to cause an event. In this case, the prompt is a text file prompt allowing the administrator to input commands into the '.dvsc' file in order to grant permissions to a second user." (See 10/23/2006 Office Action at ¶ 3.) However, as is readily understood by those in the field, and as clearly supported by the detailed description, a "prompt" necessarily includes an element of moving or inciting one to action. Thus, a prompt cannot be merely "a window on a computer screen allowing a user to input a command(s) to cause an event" (emphasis added), but must necessarily involve moving or inciting a user to input one or more commands to cause an

event. A passive ".dvsc" file, which an administrator must decide to open and modify on his or her own initiative, cannot be considered a prompt, as claimed.

In addition, the Office acknowledges that DoubleVision fails to teach replicating current contents, as claimed, and relies on Muta for this aspect of the claimed subject matter. However, Muta fails to cure the deficiencies of DoubleVision.

Muta describes systems and techniques for control of a GUI (Graphical User Interface) screen at a server in a remote location without requiring prior installment of special remote controlling software in a remote controlling machine. (See Muta at Abstract.) The Office asserts that Muta's rewriting of a GUI screen on a slave server by a window system 320 operating on the slave server constitutes replicating current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine. This contention should be overturned.

Muta clearly describes the window system 320 as operating on window messages coming from the event analyzer 315. (See Muta at col. 10, lines 42-67.) The event analyzer 315 pulls the events from the event buffer 313, and these events come from the master applet 215 at the master controller 210. (See Muta at col. 10, lines 31-41; and FIGS. 2, 3, 8, 11, and 12.) Thus,

even if Muta can be considered to replicate content from one screen to another, these screens are on different machines, not the same machine. Thus, Muta cannot be considered to teach or suggest "replicating current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine", as recited in claim 1 (emphasis added).

In response to this point, the Office has asserted that, "the fact that the screens are on different machines, is still within the scope of applicants claim language. The limitation 'running in a background' is broad and vague. Something running in the background of the UNIX machine can simply be interpreted to be another machine in the background of the UNIX machine." (See 10/23/2006 Office Action at ¶ 4.) This interpretation of the claim language should be overturned because it completely disregards the knowledge of those skilled in the computer art and the manner in which the term "background" is used in the detailed description of the present application. (See e.g., the Specification at page 2, lines 7-9, page 3, lines 10-12, and page 6, lines 19-22.)

Those of ordinary skill in the art, who are familiar with UNIX-based machines, or other similar machines that utilize a multitasking operating system, will readily understand the distinction between a background process and a foreground

process on a computing machine. For example, when a screen (or shell) on a traditional UNIX-based machine runs a foreground process, the screen waits for the foreground process to end before other processes are run in that screen. In contrast, when a screen (or shell) on a traditional UNIX-based machine runs a background process, the screen does not need to wait for the background process to end before it can run more background processes.

In view of this basic knowledge of UNIX, the claim language "replicating current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine" is clearly referring to a background process of the same computing machine from which the "current contents of a screen" are obtained. In brief, suggesting that the claim language "running in a background of the UNIX-based machine" can be interpreted as referring to a screen running on a separate machine from which the replicated contents are obtained does not make sense in view of the plain meaning of the claim language, the Specification and the knowledge of those of ordinary skill in the relevant art.

For all of the above reasons, a *prima facie* case of obviousness has not been established for any of independent claims 1, 11, 21, 27 and 31. With respect to independent claim

25, the word "background" is not specifically used, but the Office has rejected claim 25 under a similar rationale to that of the other independent claims, and the Office's position should be overturned based on similar reasoning as above.

Claim 25 recites, "replicating current contents of a display screen visible to a user on a UNIX-based device onto a new screen not visible on the display screen to the user; inserting a prompt on the new screen to a user of the UNIX-based device to grant permission for a remote device at a location remote from the UNIX-based device to control the UNIX-based device; and replacing the current contents of the display screen with the new screen, the new screen visible to the user on the UNIX-based device." (Emphasis added.) The plain meaning of this claim language is that the new screen is first not visible, and then later visible on a single UNIX-based device, which necessarily means that the new screen is at all times running on the single UNIX-based device. As described, for example, in the detailed description:

If the user terminal 104 is running in text mode, then the remote control agent 110 creates 212 a new session and the virtual console 126 in the background of the user terminal's currently active console. In other words, the remote control agent 110 creates or compiles the instructions necessary to create a new

screen for display on the user terminal 104 on the display screen 122. This new screen is created or compiled transparently to the user 102, i.e., without the user's knowledge. For example, the remote control agent 110 may open a new tty (here, a new, "virtual" terminal) in non-blocking mode and create a new stdin (a file pointer that provides access to the keyboard 114) and a new stdout and a new stderr (file pointers that write data to the display screen 122) for the new tty.

The remote control agent 110 replicates 214 the contents of the user terminal's currently active display screen 122 into the virtual console 126. To replicate the current contents of the user terminal's display screen 122, the remote control agent 110 gathers data about the display screen 122 such as the size of the display screen 122 and the cursor's position on the display screen 122. The remote control agent 110 also determines and reserves the amount of memory necessary to store the current contents of the display screen 122 to use in the replication.

The remote control agent 110 also inserts 216 a text prompt into the virtual console 126. The text prompt appears as the last text line on the virtual console 126 (although it could be anywhere) and informs the user 102 that a remote user would like to remotely control the user terminal 104. For example, the text prompt could be "Will you allow <remote user> to remote control your machine? Y/N," "Will you allow

<remote user> to transfer files? Y/N," "May another user remote control your machine?," or other similar message.

One or more text prompts may be available for display by the remote control agent 110. If two or more text prompts are available, the remote control agent 110 may choose which text prompt to display to the user 102 based on information sent in the notification from the remote terminal 108. For example, the notification could indicate that the remote user 108 desires to transfer files from the user terminal 104, in which case the remote control agent 110 would choose a text prompt such as "May <remote user> transfer files? Y/N."

After creating the virtual console 126 and inserting the text prompt, the remote control agent 110 switches 218 from the currently active display screen 122 (shown for example in FIG. 3) to the virtual console 126 (shown for example in FIG. 4). In this way, the user 102 sees on the display screen 122 what he or she was seeing before the switch to the virtual console 126 (except for the addition of the text prompt). The user 102 experiences minimal visual disruption during the switching, likely not being able to detect that any virtual terminal switching was done (except for the addition of the prompt).

(See Specification at page 6, line 19 to page 8, line 23.) In view of the above, it should be clear that the "not visible" to "visible" aspect of independent claim 25 is not rendered obvious

(in view of DoubleVision) by the Office's statement that Muta teaches replicating content from one screen to another, where these screens are on different machines, and a *prima facie* case of obviousness has not been established for independent claim 25.

Furthermore, independent claim 31 recites, "receiving a request at a first programmable data processing machine for provision of remote operational control of the first machine to a second data processing machine, the first machine being a UNIX-based machine; determining whether a user interface of the first machine is operating in a text mode; and if the first machine user interface is operating in the text mode, allocating a virtual console running in a background of the first machine, adding a prompt to the virtual console asking for authorization of the request, and switching the virtual console to be a currently active console in the first machine." (Emphasis added.) This aspect of claim 31 (determining whether a user interface of a UNIX-based machine is operating in a text mode, and performing operations based on this determination) has not yet been addressed by the Office, and no citations to the art have been provided to show where such a determination of text mode operation can be found. For at least this additional

reason, a *prima facie* case of obviousness has not been established for independent claim 31.

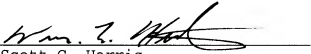
Based on the above arguments, a *prima facie* case of obviousness has also not been established for any of dependent claims 2, 3, 5-10, 12, 13, 15-20, 22-26, 28 and 32. Edwards fails to cure the deficiencies of DoubleVision and Muta. Thus, for all of the above reasons, it is respectfully requested that both Ground of Rejection I and Ground of Rejection II should be overturned.

Please apply the brief fee in the amount of \$500, and any other necessary charges or credits, to Deposit Account No. 06-1050.

Respectfully submitted,

Date: March 23, 2007

for


Scott C. Harris
Reg. No. 32,030
Attorney for Intel Corporation

Fish & Richardson P.C.
PTO Customer No. **20985**
Telephone: (858) 678-5070
Facsimile: (858) 678-5099

WILLIAM E. HUNTER
REG. NO 47,671

Appendix of Claims

1. A method comprising:

replicating current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine;

prompting a first user at the UNIX-based machine for permission for a second user at a machine remotely-located from the UNIX-based machine to control the UNIX-based machine; and

if the first user grants permission, enabling the second user to use the UNIX-based machine through the machine remotely-located from the UNIX-based machine.

2. The method of claim 1 in which the prompting comprises making the prompt known to the first user by displaying information on a display of the UNIX-based machine.

3. The method of claim 1 in which the second user uses the UNIX-based machine through the machine remotely-located from the UNIX-based machine as if the second user was directly using the UNIX-based machine.

4. (cancelled)

5. The method of claim 1 further comprising adding to the new screen a prompt that asks the first user for the permission.

6. The method of claim 1 further comprising replacing the current contents of the screen on the UNIX-based machine with the new screen.

7. The method of claim 1 in which the using of the UNIX-based machine includes issuing text commands to the UNIX-based machine from the machine remotely-located from the UNIX-based machine.

8. The method of claim 1 further comprising, if the first user does not grant permission, preventing the second user from using the UNIX-based machine through the machine remotely-located from the UNIX-based machine.

9. The method of claim 1 further comprising, if the first user at the UNIX-based machine does not respond to the prompting within a certain threshold time, enabling by default the second user to use the UNIX-based machine.

10. The method of claim 1 in which the prompting is text-based.

11. An article comprising:
a machine-readable medium which stores machine-executable instructions, the instructions causing a machine to:

replicate current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine;

prompt a first user at the UNIX-based machine for permission for a second user at a machine remotely-located from the UNIX-based machine to control the UNIX-based machine; and

if the first user grants permission, enable the second user to use the UNIX-based machine through the machine remotely-located from the UNIX-based machine.

12. The article of claim 11 in which the prompting includes making the prompt known to the first user by displaying information on a display of the UNIX-based machine.

13. The article of claim 11 in which the second user uses the UNIX-based machine through the machine remotely-located from the UNIX-based machine as if the second user was directly using the UNIX-based machine.

14. (cancelled)

15. The article of claim 11 further causing a machine to add to the new screen a prompt that asks the first user for the permission.

16. The article of claim 11 further causing a machine to replace the current contents of the screen on the UNIX-based machine with the new screen.

17. The article of claim 11 in which the using of the UNIX-based machine includes issuing text commands to the UNIX-based machine from the machine remotely-located from the UNIX-based machine.

18. The article of claim 11 further causing a machine to, if the first user does not grant permission, prevent the second user from using the UNIX-based machine through the machine remotely-located from the UNIX-based machine.

19. The article of claim 11 further causing a machine to, if the first user at the UNIX-based machine does not respond to the prompting within a certain threshold time, enable by default the second user to use the UNIX-based machine.

20. The article of claim 11 in which the prompting is text-based.

21. A system comprising:

a first device configured to run UNIX; and

a mechanism accessible by the first device and configured

to run a process on the first device transparently to a user of the first device, the process configured to replicate current contents of a screen on the first device onto a new screen running in a background of the first device and prompt the user of the first device for permission for a remote user at a second device at a location remote from the first device to input instructions to the first device from the second device.

22. The system of claim 21 in which the process is also configured to, if the user of the first device grants permission, enable the remote user to use the first device through the second device as if the remote user was directly using the first device.

23. The system of claim 21 further comprising a second mechanism accessible by the second device and configured to notify the first device when the remote user desires to input instructions to the first device from the second device.

24. The system of claim 21 in which the process is also configured to continuously run on the first device.

25. A method comprising:
replicating current contents of a display screen visible to a user on a UNIX-based device onto a new screen not visible on

the display screen to the user;

inserting a prompt on the new screen to a user of the UNIX-based device to grant permission for a remote device at a location remote from the UNIX-based device to control the UNIX-based device; and

replacing the current contents of the display screen with the new screen, the new screen visible to the user on the UNIX-based device.

26. The method of claim 25 further comprising, after the user responds to the prompt, returning the UNIX-based device back to the current contents of the display screen.

27. A method comprising:

replicating current contents of a screen on a UNIX-based machine onto a new screen running in a background of the UNIX-based machine;

adding to the new screen a text prompt prompting a first user at the UNIX-based machine for permission for a second user at a machine remotely-located from the UNIX-based machine to control the UNIX-based machine; and

replacing the current contents of the screen with the new screen.

28. The method of claim 27 further comprising determining if the second user may control the UNIX-based machine based on a response to the text prompt by the first user.

29-30. (cancelled)

31. A machine-implemented method comprising:

receiving a request at a first programmable data processing machine for provision of remote operational control of the first machine to a second data processing machine, the first machine being a UNIX-based machine;

determining whether a user interface of the first machine is operating in a text mode; and

if the first machine user interface is operating in the text mode, allocating a virtual console running in a background of the first machine, adding a prompt to the virtual console asking for authorization of the request, and switching the virtual console to be a currently active console in the first machine.

32. The method of claim 31, further comprising:

if the first machine user interface is operating in the text mode, replicating contents of an originally active console to the virtual console before adding the prompt and switching

the virtual console to be the currently active console; and
after receiving an input or a time out, switching the
originally active console to then be the currently active
console, and deallocating the virtual console.

Applicant : Eric D. Fagerburg et al.
Serial No.: 09/752,100
Filed: December 28, 2000
Page : 25 of 26

Attorney's Docket No.: 10559-322001
P9683

Intel Corporation

Evidence Appendix

None.

Applicant : Eric D. Fagerburg et al.
Serial No.: 09/752,100
Filed: December 28, 2000
Page : 26 of 26

Attorney's Docket No.: 10559-322001
P9683
Intel Corporation

Related Proceedings Appendix

None.